

主题：关于四面体程序的分析与小结

旧版本简介：四面体程序的原有版本是在我去年写的程序以及李昕的四面体光线投射程序基础上，由我和陆少超共同完成的。该程序实现传统的采用预积分技术的四面体数据光线投射算法。但是有一个问题：即不能完整处理凹四面体数据或者内部有洞的数据，亦即不能处理“光线再入”问题。因此往往以 **bluntFin** 或者 **comb** 等数据来作为本程序的展示数据。以下简称此版本为基础版本。（注：新数据为 **spx** 数据，内部有洞）

新版本由来：由于原先论文理论的错误，新理论按鲁佳硕士论文的方法高次重构了整个 **mesh** 的数据场，并且增加了四面体的中间结点数据。由此，必须修改基础版本的相应代码。当前程序所有代码在 **source** 文件夹中，**sln** 和工程文件在 **build** 文件夹中，数据和可执行文件在 **bin** 文件夹中。

程序修改：程序修改主要有四个部分。

应用程序中修改部分全部用 **DZY_CORRECT** 预编译指令进行控制

GPU 代码由 **useQuadraticTetra** 预编译指令控制

修改的代码主要有以下四处

1. **Precomp.h** 文件中。增加所有关于中间结点的数据结构定义和成员定义。包括中间结点索引，数据场数组，中间结点数据场纹理 ID 等。
2. **Precomp.cpp** 文件中，这是最主要的部分。内含对中间结点数据场的处理，包括从文件中读取数据值，构建 **mesh**，生成中间节点三维纹理等。
3. **Main.cpp** 文件中，主要是增加由 CPU 向 GPU 传数据传纹理部分代码。
4. **SecondFragmentShader.h** 文件中，本文件是主要执行渲染的文件，所有光线投射的核心渲染代码。增加部分主要功能是从中间节点纹理中拾取纹理数据，将获取的中间节点数据与角节点数据按论文的公式重构出正确的数据场，然后渲染。

需要解决的主要问题：在基础版本上，如何正确的加入中间节点数据信息。因为每个中间节点是按一定的顺序排列的，在原有程序基础上正确的插入这些信息而不影响任何一个环节就是需要解决的核心问题。

当前的状态：在构建中间结点数据信息到原有程序的过程中，某些环节被错误的影响到，可能是原有的邻接关系的信息被破坏了，一直没能准确的捕获错误源。涉及的代码是 **precomp.cpp** 中 **buildTexture** 函数。在当前程序中，若关闭 **DZY_CORRECT** 和 **useQuadraticTetra** 指令，则就是原来的基础版本程序，绘制没有问题。开启以后，渲染的结果是不正确的。在包含当前最新版本的 **precomp.cpp** 文件下运行程序时，会有运行时刻错误，这是今天刚写的，还没有正确调出来，但是改写的思路在代码中可以看出来。这之前的上一个版本的 **precomp.cpp** 文件在 **Newfolder** 中，只要拷贝它覆盖 **source** 文件夹下的同名文件即可，在包含该文件的情形下运行程序，若不开启上述两个命令，则运行也是基础版本的情形，没有问题。若开启，可以得到绘制结果，但是是不正确的结果。

运行数据：在 main.cpp 文件中第 80 和 83 行可以修改需要运行的数据。当前是 zaoshuju.off & zaoshuju.moff 这两个数据是人工生成的测试数据 仅仅包含 2 个四面体，用于测试和调试。若要运行常规数据，改成 dzy.off & dzy.moff 即可。